Modelling and Control of a Quadrotor UAV System using Non-Linear PID Methods

K. Malkow
k.malkow@student.tue.nl, ID 1266152
Eindhoven University of Technology
Eindhoven, Netherlands

***Abstract*—**

I. Introduction

QUADROTOR Unmanned Aerial Vehicles (UAVs) are vehicles which are propelled by four rotors, two of which rotate in a clockwise direction, and two of which rotate in an anti-clockwise direction. These vehicles come in different sizes, are relatively easy to implement, are low cost, and are capable of vertical take-off and landing. All of which contribute to the fact that these vehicles are able to be applied to a wide range of uses, which have led to a spike of interest in such UAVs in recent years. Several such applications include agricultural applications [1]–[3], applications in rescue [4], [5], and applications in payload transportation [6]–[8]. These types of applications require quadrotor UAVs to execute tasks, which are only possible with a properly functioning control system.

A simple, robust, and proven way to control quadrotor UAVs is using linear PID (LPID) controllers [9]–[11]. However, as is shown in [12], LPID controllers have the disadvantage that these are not able to perform well in places where non-linearities exist, such as in translation or in complex trajectories, where some combination of translation and ascending/descending is present. These LPID controllers may perform well around a linearization point (such as in hovering), however, the further away the quadrotor UAV gets from such a linearization point, the worse the performance of the LPID controller gets. Therefore, non-linear controllers, which are able to deal with such non-linearities, are required for better performance. In general, the performance of non-linear controllers is superior to that of LPID controllers. In particular, non-linear PID (NLPID) controllers are of interest, since a direct comparison can be drawn to LPID controllers. Implementations of such NLPID controllers to different applications can be observed in [12]–[17].

While this literature may include functioning NLPID controller implementations as well as comparisons with LPID controllers, little research has been done into applying such controllers to a quadrotor UAV system. Furthermore, in this literature, little to no attention has been payed to tuning methods or even simple tuning methods for NLPID controllers. Often, the parameters for the NLPID controllers are obtained by solving a non-linear optimization problem through the use of the Genetic Algorithm as is done in [11], [12].

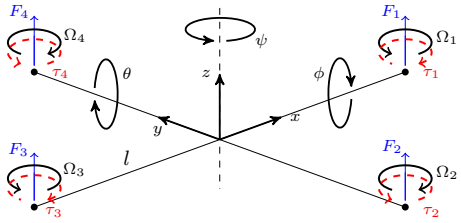

Fig. 1. Coordinate system for the quadrotor UAV plant model

This paper aims to present a quadrotor UAV model in Newton-Euler formulation, the design and implementation of a NLPID controller on this quadrotor UAV model using a simple tuning method, as well as a comparison with a LPID controller. The goal of which is to investigate whether the performance of the NLPID controller, which uses a simple tuning method, is better than the performance of a LPID controller when applied to a quadrotor UAV system for complex trajectories. To accomplish this, this paper mainly utilizes the work presented in [12], in which a NLPID controller implementation on a quadrotor UAV system is given. Furthermore, MATLAB/Simulink is used for the software implementation.

This paper is structured as follows: Section II describes the quadrotor UAV plant model, which includes the dynamics of the quadrotor UAV system as well as the dynamics and intricacies of the actuators.

II. Quadrotor Plant Model

In this section, a mathematical model of the quadrotor UAV plant characterising the dynamics of this system is presented. This is required, because to control a system the characteristics of the system must be modelled. The coordinate system used for this model is shown in Fig. 1.

An overview of the architecture implemented in MATLAB/Simulink is shown in Fig. 2. This architecture comprises two main parts, namely the controller and the quadrotor plant model. The quadrotor plant model, which is divided into three parts, namely the conversions, the actuator dynamics, and the system dynamics, is considered first.

A. System Dynamics

In this part, the system dynamics, which comprise the equations of motion, the state space model, the linearized state

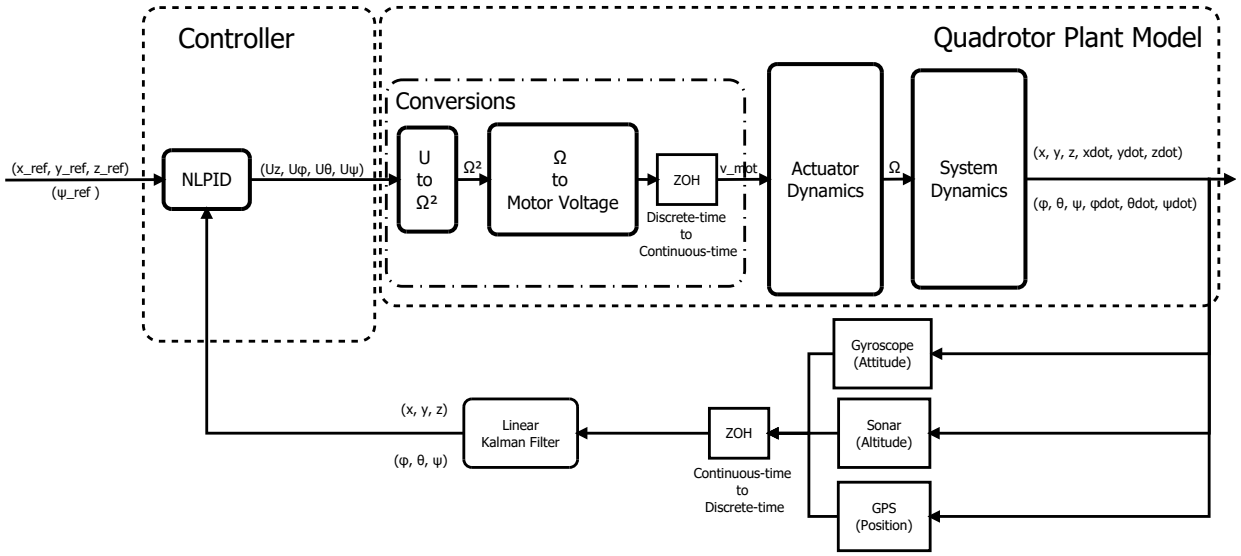

Fig. 2. Overview of architecture implemented in MATLAB/Simulink

space model, and the linearized transfer functions, is presented. The model used for the system dynamics is based on the model given in [12], [18], in which, among others, a rigid body assumption with a symmetrical structure is made and the ground effect is neglected. The former allows for the aeroelasticity to be neglected due to the small size of the vehicle [19]. If the translation and rotation in terms of position in the inertial frame of reference is given by the vector $[x \;\; y \;\; z \;\; \phi \;\; \theta \;\; \psi]^T$ and if the translation and rotation in terms of rates in the body frame of reference is given by the vector $[u \;\; v \;\; w \;\; p \;\; q \;\; r]^T$, then the equations of motion are given by

$$\begin{cases} \dot{x} = u[c(\psi)c(\theta)] + v[c(\psi)s(\phi)s(\theta) - c(\phi)s(\psi)] \\ +w[s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta)] \\ \dot{y} = u[c(\theta)s(\psi)] + v[s(\phi)s(\psi)s(\theta) + c(\phi)c(\psi)] \\ +w[c(\phi)s(\psi)s(\theta) - c(\psi)s(\phi)] \\ \dot{z} = -u[s(\theta)] + v[c(\theta)s(\phi)] + w[c(\phi)c(\theta)] \\ \dot{u} = vr - wq - g[s(\theta)] \\ \dot{v} = -ur + wp + g[s(\phi)c(\theta)] \\ \dot{w} = uq - vp + g[c(\phi)c(\theta)] - \frac{f_t}{m} \\ \dot{\phi} = p + q[s(\phi)t(\theta)] + r[c(\phi)t(\theta)] \\ \dot{\theta} = q[c(\phi)] - r[s(\phi)] \\ \dot{\psi} = q[\frac{s(\phi)}{c(\theta)}] + r[\frac{c(\phi)}{c(\theta)}] \\ \dot{p} = rq[\frac{I_y - I_z}{I_x}] + \frac{\tau_x}{I_x} \\ \dot{q} = pr[\frac{I_z - I_x}{I_y}] + \frac{\tau_y}{I_y} \\ \dot{r} = pq[\frac{I_x - I_y}{I_z}] + \frac{\tau_z}{I_z}, \end{cases} \tag{1}$$

where g is the gravitational acceleration, m is the mass of the quadrotor UAV, I_x is the moment of inertia along the x-axis, I_y is the moment of inertia along the y-axis, and I_z is the moment of inertia along the z-axis. Furthermore, f_t is the total thrust force which controls the altitude z, τ_x is the torque which controls the roll (ϕ), τ_y is the torque which controls the pitch (θ), and τ_z is the torque which controls the yaw (ψ), all

TABLE I
QUADROTOR UAV MODEL PARAMETERS

Symbol	Variable	Value	Unit
m	Quadrotor mass	5	kg
I_x	Moment of inertia along x-axis	0.03	kgm^2
I_y	Moment of inertia along y-axis	0.03	kgm^2
I_z	Moment of inertia along z-axis	0.03	kgm^2
b	Aerodynamic force coefficient	1	-
d	Aerodynamic torque coefficient	0.6	-
J_r	Rotor inertia	0.01	kgm^2
l	Rotor to center length	0.5	m

of which act as input to the system dynamics and are given by

$$\begin{cases} f_t = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ \tau_x = bl(\Omega_4^2 - \Omega_2^2) \\ \tau_y = bl(\Omega_3^2 - \Omega_1^2) \\ \tau_z = d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2), \end{cases} \tag{2}$$

where b is the aerodynamic force coefficient, l is the length from each rotor to the centre of the quadrotor UAV, d is the aerodynamic torque coefficient, and Ω_i where $i \in \{1, ..., 4\}$ is the rotor speed of each corresponding motor. The derivation of which is given in Appendix A. The values of the corresponding variables can be seen in Table I [20]. Please note that s(α) = $\sin(\alpha)$, c(α) = $\cos(\alpha)$, and t(α) = $\tan(\alpha)$ and that the influence of aerodynamic drag forces and torques, which occur due to air friction, has been assumed to be negligible to reduce the complexity of the model. Furthermore, disturbances, which mainly occur due to wind, are also neglected since quadrotor UAVs tend to be rarely flown in environments with strong winds and hence strong disturbances.

While the equations of motion given in (1) may describe the motion of the 6-DOF quadrotor UAV, these equations of motion are too complex to gain reasonable insight into the motion of the quadrotor UAV and are hence not suitable for control design [21].

To get appropriate insight, the equations of motion are re-written into an inertial frame of reference. This is done using the methods outlined in [11], [18], [21], [22] in which the Euler angle convention and Newton's law are used to obtain the Newton-Euler formulation. Using the Euler angle convention, the rotation from the body frame of reference to the inertial frame of reference is described by the rotation matrix $\mathbf{R}$ which is given by

$$\mathbf{R} = \begin{bmatrix} c(\theta)c(\psi) & s(\phi)s(\theta)c(\psi) - c(\phi)s(\psi) & c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi) \\ c(\theta)s(\psi) & s(\phi)s(\theta)s(\psi) + c(\phi)c(\psi) & c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi) \\ -s(\theta) & s(\phi)c(\theta) & c(\phi)c(\theta) \end{bmatrix}. \quad (3)$$

Furthermore, the angular rates in the body frame of reference $[p \ q \ r]^T$ are related to the angular rates in the inertial frame of reference $[\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$ by

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s(\phi)t(\theta) & c(\phi)t(\theta) \\ 0 & c(\phi) & -s(\phi) \\ 0 & \frac{s(\phi)}{c(\theta)} & \frac{c(\phi)}{c(\theta)} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \quad (4)$$

Using Newton's law, which is given by

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \mathbf{R} \begin{bmatrix} 0 \\ 0 \\ f_t \end{bmatrix}, \quad (5)$$

the equations of motion in Newton-Euler formulation become

$$\begin{cases} \ddot{x} = \frac{f_t}{m}(c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi)) \\ \ddot{y} = \frac{f_t}{m}(c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi)) \\ \ddot{z} = \frac{f_t}{m}(c(\phi)c(\theta)) - g \\ \ddot{\phi} = \dot{\psi}\dot{\theta}[\frac{I_y - I_z}{I_x}] + \dot{\theta}\frac{J_r}{I_x}\Omega_r + \frac{\tau_x}{I_x} \\ \ddot{\theta} = \dot{\phi}\dot{\psi}[\frac{I_z - I_x}{I_y}] - \dot{\phi}\frac{J_r}{I_y}\Omega_r + \frac{\tau_y}{I_y} \\ \ddot{\psi} = \dot{\phi}\dot{\theta}[\frac{I_x - I_y}{I_z}] + \frac{\tau_z}{I_z}, \end{cases} \quad (6)$$

where J_r is the rotor inertia and Ω_r is the residual overall rotor speed given by

$$\Omega_r = \Omega_2 + \Omega_4 - \Omega_1 - \Omega_3. \quad (7)$$

To obtain the equations of motion given in (6), it should be noted that it is assumed that the quadrotor operates around hovering, which allows for making the assumption that the movement in ϕ and θ is small. This permits directly relating $[\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$ to $[p \ q \ r]^T$. Furthermore, both J_r and Ω_r are used to represent the propeller gyroscopic effect for roll $\dot{\theta}\frac{J_r}{I_x}\Omega_r$ and for pitch $\dot{\phi}\frac{J_r}{I_y}\Omega_r$ [9], [10], [23] and were added to make the model more accurate.

By re-writing the equations of motion given in (6), a state-space model $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ with state vector $\mathbf{x}$ given by

$$\mathbf{x} = [x \ \dot{x} \ y \ \dot{y} \ z \ \dot{z} \ \phi \ \dot{\phi} \ \theta \ \dot{\theta} \ \psi \ \dot{\psi}]^T, \quad (8)$$

and control input vector $\mathbf{u}$ given by

$$\mathbf{u} = [u_z \ u_\phi \ u_\theta \ u_\psi]^T, \quad (9)$$

where

$$u_z = f_t, \ u_\phi = \tau_x, \ u_\theta = \tau_y, \ u_\psi = \tau_z, \quad (10)$$

is constructed.

The resulting state space model is given by

$$\dot{\mathbf{x}} = \begin{cases} \dot{x} \\ \frac{u_z}{m}(c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi)) \\ \dot{y} \\ \frac{u_z}{m}(c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi)) \\ \dot{z} \\ \frac{u_z}{m}(c(\phi)c(\theta)) - g \\ \dot{\phi} \\ \dot{\psi}\dot{\theta}[\frac{I_y - I_z}{I_x}] + \dot{\theta}\frac{J_r}{I_x}\Omega_r + \frac{u_\phi}{I_x} \\ \dot{\theta} \\ \dot{\phi}\dot{\psi}[\frac{I_z - I_x}{I_y}] - \dot{\phi}\frac{J_r}{I_y}\Omega_r + \frac{u_\theta}{I_y} \\ \dot{\psi} \\ \dot{\phi}\dot{\theta}[\frac{I_x - I_y}{I_z}] + \frac{u_\psi}{I_z}. \end{cases} \quad (11)$$

To obtain a linearized model of the quadrotor UAV plant, the methods used in [12], [18] are used. This linearized model is required for the control design and to obtain it a linearization is applied to the state space model given in (11) around an equilibrium point which corresponds to hovering given by

$$\mathbf{x_0} = [x \ 0 \ y \ 0 \ z \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T. \quad (12)$$

To obtain this, the control input vector from (9) is given by the constant input value

$$\mathbf{u_0} = [mg \ 0 \ 0 \ 0]^T. \quad (13)$$

Using (12) and (13), the linearized model is given by

$$\dot{\mathbf{x}} \approx \mathbf{f}(\mathbf{x_0}, \mathbf{u_0}) + \mathbf{A} \cdot (\mathbf{x} - \mathbf{x_0}) + \mathbf{B} \cdot (\mathbf{u} - \mathbf{u_0}), \quad (14)$$

where $\mathbf{A}$ and $\mathbf{B}$ are matrices given by

$$\mathbf{A} = \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}}\Big|_{\substack{\mathbf{x}=\mathbf{x_0} \\ \mathbf{u}=\mathbf{u_0}}}, \ \mathbf{B} = \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}}\Big|_{\substack{\mathbf{x}=\mathbf{x_0} \\ \mathbf{u}=\mathbf{u_0}}}, \quad (15)$$

the values of which can be found in Appendix B. It should be noted, that since a hovering situation is considered and movement in ϕ and θ is therefore assumed to be small, the contribution by the gyroscopic effects for roll and pitch are negligible and are hence not included in the control design [24].

The linearized state space model of the quadrotor UAV plant is then given by

$$\dot{\mathbf{x}} = \begin{cases} \dot{x} \\ u_x \\ \dot{y} \\ u_y \\ \dot{z} \\ \frac{1}{m}(u_z - mg) \\ \dot{\phi} \\ \frac{u_\phi}{I_x} \\ \dot{\theta} \\ \frac{u_\theta}{I_y} \\ \dot{\psi} \\ \frac{u_\psi}{I_z}, \end{cases} \tag{16}$$

where x-movement and y-movement are forced by control inputs u_x and u_y, respectively, which will be derived later in this paper. It is important to note that this formulation of x-movement and y-movement is different from the corresponding formulation given in (26) of Appendix C.

From the linearized state space model given in (16), the corresponding transfer functions which are required for the control design can be derived and written into the Laplace domain [25]. To obtain these, $\frac{1}{m}(u_z - mg)$ is rewritten to $\frac{1}{m}\Delta u_z$. Since the linearized model is obtained around $u_z = mg$, if $u_z > mg$, Δu_z is positive and the resulting motion is in an upwards direction. If $u_z < mg$, Δu_z is negative and the resulting motion is in a downwards direction. This was implemented in MATLAB/Simulink by adding a constant mg to the control input u_z. The resulting transfer functions in the Laplace domain are given by

$$\begin{aligned} \frac{X(s)}{U_x(s)} &= \frac{1}{s^2} \\ \frac{Y(s)}{U_y(s)} &= \frac{1}{s^2} \\ \frac{Z(s)}{\Delta U_z(s)} &= \frac{1}{m}\frac{1}{s^2} \\ \frac{\Phi(s)}{U_\phi(s)} &= \frac{1}{I_x}\frac{1}{s^2} \\ \frac{\Theta(s)}{U_\theta(s)} &= \frac{1}{I_y}\frac{1}{s^2} \\ \frac{\Psi(s)}{U_\psi(s)} &= \frac{1}{I_z}\frac{1}{s^2}. \end{aligned} \tag{17}$$

From (17), it can be seen that all transfer functions are of a second order type, in which both poles are situated at the origin, making the quadrotor UAV plant inherently unstable.

B. Actuator Dynamics

C. Conversions

III. Controller

The controller structure is shown in Fig. 3. The structure is the same for both the LPID and NLPID implementations. The utilized LPID controller as well as linear Kalman filter were both obtained from [20].

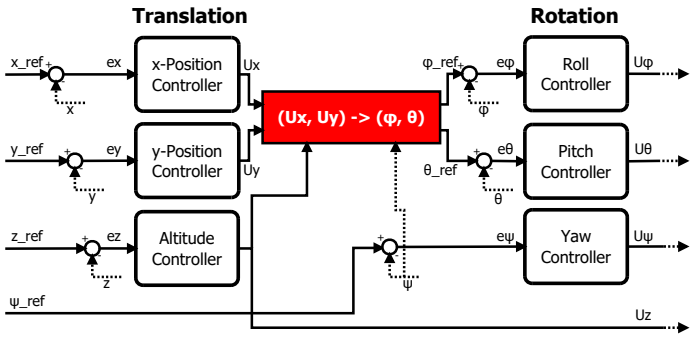

Fig. 3. Overview of controller structure implemented in MATLAB/Simulink

IV. Reference Signal Inputs for Simulation

V. Results and Discussion

VI. Conclusion and Recommendations

References

[1] J. Navia, I. Mondragon, D. Patino, and J. Colorado, "Multispectral mapping in agriculture: Terrain mosaic using an autonomous quadcopter uav," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2016, pp. 1351–1358.

[2] V. H. Andaluz, E. López, D. Manobanda, F. Guamushig, F. Chicaiza, J. S. Sánchez, D. Rivas, F. Pérez, C. Sánchez, and V. Morales, "Nonlinear controller of quadcopters for agricultural monitoring," in *Advances in Visual Computing*, G. Bebis, R. Boyle, B. Parvin, D. Koracin, I. Pavlidis, R. Feris, T. McGraw, M. Elendt, R. Kopper, E. Ragan, Z. Ye, and G. Weber, Eds. Cham: Springer International Publishing, 2015, pp. 476–487.

[3] S. M. Meivel, A. Professor, R. Maguteeswaran, N. B. Gandhiraj, and G. Srinivasan, "Quadcopter UAV Based Fertilizer and Pesticide Spraying System," *©International Academic Research Journal of Engineering Sciences*, vol. 1, no. 1, pp. 2414–6242, 2016.

[4] R. D. Angelo and R. Levin, "Design of an Autonomous Quadrotor UAV for Urban Search and Rescue," p. 104, 2011.

[5] X. Liang, G. Chen, J. Wang, Z. Bi, and P. Sun, "An adaptive control system for variable mass quad-rotor UAV involved in rescue missions," *International Journal of Simulation: Systems, Science and Technology*, vol. 17, no. 29, pp. 22.1–22.7, 2016.

[6] S. Yang and B. Xian, "Energy-based nonlinear adaptive control design for the quadrotor uav system with a suspended payload," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 3, pp. 2054–2064, 2020.

[7] S. Dai, T. Lee, and D. S. Bernstein, "Adaptive control of a quadrotor uav transporting a cable-suspended load with unknown mass," in *53rd IEEE Conference on Decision and Control*, 2014, pp. 6149–6154.

[8] F. A. Goodarzi, D. Lee, and T. Lee, "Geometric control of a quadrotor UAV transporting a payload connected via flexible cable," *International Journal of Control, Automation and Systems*, vol. 13, no. 6, pp. 1486–1498, 2015.

[9] S. Bouabdallah, "Design and Control of Quadrotors With Application To Autonomous Flying," *École Polytechnique Fédérale De Lausanne, À La Faculté Des Sciences Et Techniques De L'Ingénieur*, vol. 3727, no. 3727, p. 61, 2007.

[10] D. Kotarski, Z. Benic, and M. Krznar, "Control Design for Unmanned Aerial Vehicles with Four Rotors," *Interdisciplinary Description of Complex Systems*, vol. 14, no. 2, pp. 236–245, 2016.

[11] M. K. Habib, W. G. A. Abdelaal, and M. S. Saad, "Dynamic Modeling and Control of a Quadrotor Using Linear and Nonlinear Approaches," 2014. [Online]. Available: http://dar.aucegypt.edu/handle/10526/3965

[12] A. A. Najm and I. K. Ibraheem, "Nonlinear PID controller design for a 6-DOF UAV quadrotor system," *Engineering Science and Technology, an International Journal*, vol. 22, no. 4, pp. 1087–1097, 2019. [Online]. Available: https://doi.org/10.1016/j.jestch.2019.02.005

[13] W. Riyadh and I. Kasim, "From PID to Nonlinear State Error Feedback Controller," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 1, pp. 312–322, 2017.

[14] Huanpao Huang, "Nonlinear pid controller and its applications in power plants," in *Proceedings. International Conference on Power System Technology*, vol. 3, Oct 2002, pp. 1513–1517 vol.3.

[15] J.-Q. Han, "Nonlinear design methods for control systems," *IFAC Proceedings Volumes*, vol. 32, no. 2, pp. 1531–1536, 1999. [Online]. Available: http://dx.doi.org/10.1016/S1474-6670(17)56259-X
[16] Y. X. Su, B. Y. Duan, and C. H. Zheng, "Nonlinear PID control of a six-DOF parallel manipulator," *IEE Proceedings: Control Theory and Applications*, vol. 151, no. 1, pp. 95–102, 2004.
[17] Y. X. Su, D. Sun, and B. Y. Duan, "Design of an enhanced nonlinear PID controller," *Mechatronics*, vol. 15, no. 8, pp. 1005–1024, 2005.
[18] F. Sabatino, "Quadrotor control: modeling, nonlinear control design, and simulation," no. June, 2015.
[19] S. Formentin and M. Lovera, "Flatness-based control of a quadrotor helicopter via feedforward linearization," *Proceedings of the IEEE Conference on Decision and Control*, pp. 6171–6176, 2011.
[20] S. C. M. Mennen, "Quadcopter Modelling, Control and State Estimation," Bachelor End Project at Eindhoven University of Technology, please contact m.lazar@tue.nl or the author for this document.
[21] R. W. Beard, "Quadrotor Dynamics and Control," *Brigham Young University*, 2008. [Online]. Available: https://scholarsarchive.byu.edu/facpub/1325
[22] C. Powers, D. Mellinger, and V. Kumar, *Quadrotor Kinematics and Dynamics*, 01 2015, pp. 307–328.
[23] S. Bouabdallah and R. Siegwart, "Full control of a quadrotor," *IEEE International Conference on Intelligent Robots and Systems*, pp. 153–158, 2007.
[24] H. Bolandi, M. Rezaei, R. Mohsenipour, H. Nemati, and S. M. Smailzadeh, "Attitude Control of a Quadrotor with Optimized PID Controller," *Intelligent Control and Automation*, vol. 4, no. 3, pp. 335–342, 2013.
[25] G. F. Franklin, D. J. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 7th ed. Pearson, 2015, ch. 3, pp. 121–125.

APPENDIX A
AERODYNAMIC FORCES AND TORQUES ACTING ON QUADROTOR UAV SYSTEM

The derivation of the aerodynamic forces and torques which act on the quadrotor UAV system is based on the methods highlighted in [11]. The aerodynamic forces are given by

$$F_i = b\Omega_i^2 \quad \text{for} \quad i \in \{1, ..., 4\}, \tag{18}$$

and the aerodynamic torques are given by

$$\tau_i = d\Omega_i^2 \quad \text{for} \quad i \in \{1, ..., 4\}. \tag{19}$$

By observing the aerodynamic forces and torques shown in Fig. 1, f_t, τ_x, τ_y, and τ_z can be found. To obtain the total thrust force f_t, the thrust force produced by each rotor given in (18) is added together to obtain the total thrust given by

$$f_t = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2), \tag{20}$$

where it should be noted that this force is positive since the thrust is upwards and the positive z-axis is also pointing upwards.

To obtain the torque taken about the x-axis τ_x, the right hand rule is applied. This results in a positive torque generated by multiplying F_4 with the arm l and a negative torque generated in a similar manner by F_2. Therefore, the total torque taken about the x-axis is given by

$$\tau_x = F_4 l - F_2 l = (b\Omega_4^2)l - (b\Omega_2^2)l = bl(\Omega_4^2 - \Omega_2^2). \tag{21}$$

Similarly, to obtain the torque taken about the y-axis τ_y, the right hand rule is also applied. This results in a positive torque generated by multiplying F_3 with the arm l and a negative torque generated in a similar manner by F_1. Therefore, the total torque taken about the y-axis is given by

$$\tau_y = F_3 l - F_1 l = (b\Omega_3^2)l - (b\Omega_1^2)l = bl(\Omega_3^2 - \Omega_1^2). \tag{22}$$

To obtain the torque taken about the z-axis τ_z, the thrust forces generated by each rotor is not considered. Instead, the torque generated by the rotation of each rotor given in (19) is considered and the right hand rule is applied to the z-axis of the coordinate system. This results in the total torque taken about the z-axis, which is given by

$$\begin{aligned} \tau_z &= \tau_2 + \tau_4 - \tau_1 - \tau_3 \\ &= (d\Omega_2^2) + (d\Omega_4^2) - (d\Omega_1^2) - (d\Omega_3^2) \\ &= d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2). \end{aligned} \tag{23}$$

APPENDIX B
LINEARIZED MODEL MATRICES

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -g & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{24}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{I_z} \end{bmatrix} \tag{25}$$

APPENDIX C
FULL LINEARIZED STATE SPACE

$$\dot{\mathbf{x}} = \begin{cases} \dot{x} \\ g\theta \\ \dot{y} \\ -g\phi \\ \dot{z} \\ \frac{1}{m}(u_z - mg) \\ \dot{\phi} \\ \frac{u_\phi}{I_x} \\ \dot{\theta} \\ \frac{u_\theta}{I_y} \\ \dot{\psi} \\ \frac{u_\psi}{I_z} \end{cases} \tag{26}$$